

SERVUS!

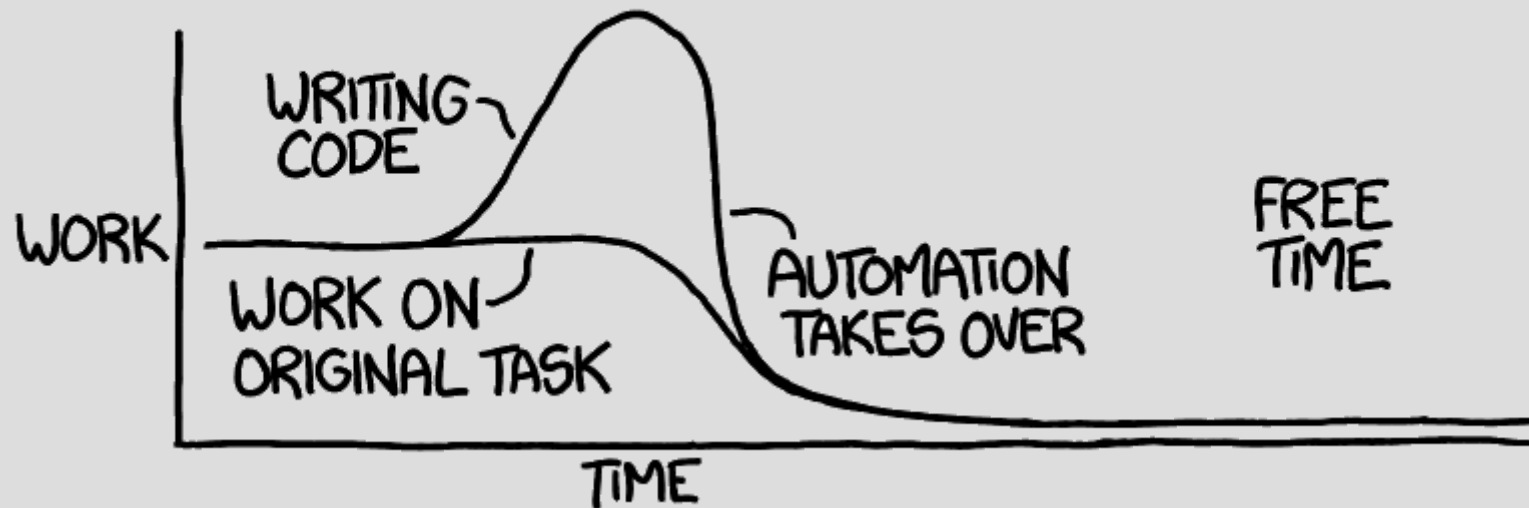
HELLO!

ПРИВЕТ!

ABOUT ME

I'M LAZY

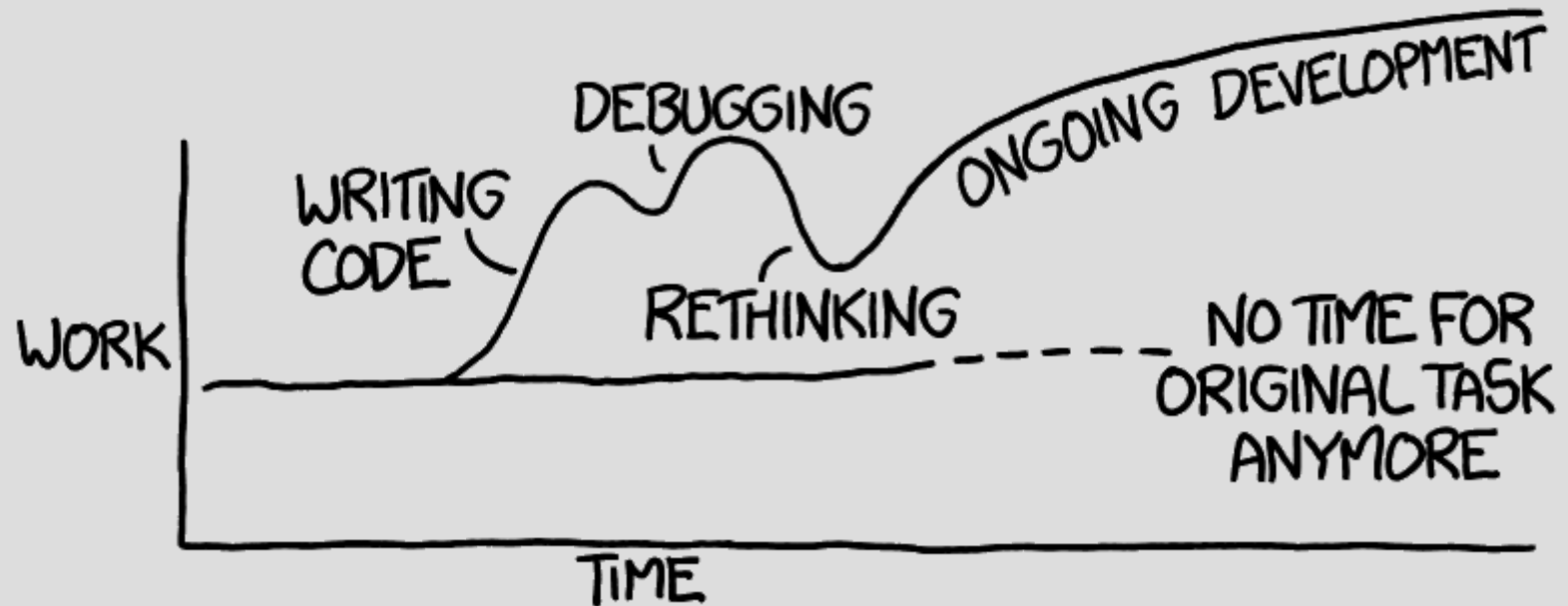
AUTOMATION



Source: <https://xkcd.com/1319/>

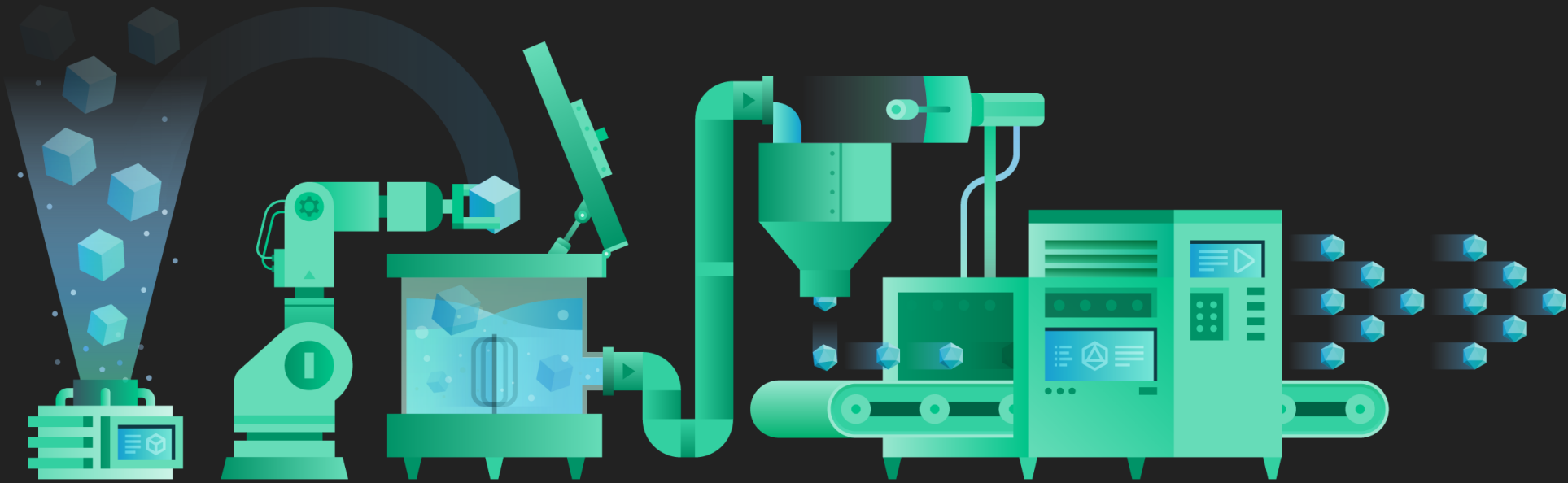
- ☛ But being lazy makes me care a lot about automation
- ☛ Taking repetitive tasks, and putting them into some kind of algorithm, so that I can relax again

AUTOMATION



Source: <https://xkcd.com/1319/>

BUILD AUTOMATION



Source: <https://gradle.org>

- ⚡ Process of automating the creation of a software build
- ⚡ Build steps like Cleaning, Compiling, Packaging
- ⚡ Reproducible => eliminate chance of human error

Branch: master
roslyn / build / scripts /

Create new file
Upload files
Find file
History

genlu
Address comments

Latest commit cf6dcdc 12 days ago

..

LoadNuGetInfo.cmd	Fix the packing issue	2 months ago
LoadNuGetInfo.ps1	Add LoadNuGetInfo.ps1 for powershell as well	a year ago
README.md	Separate out the scripts directory	2 years ago
Run-TestsWithProcessWatchdog.ps1	Integrate ProcessWatchdog into future branch	2 years ago
build-utils.ps1	Fix bug in build-utils Ensure-SdkInPathAndData	a month ago
build-utils.sh	Quote variable/array expansions in shell scripts	2 months ago
build.ps1	Address comments	12 days ago
check-toolset-insertion.ps1	Ignore DSRN when checking VSIX content	7 months ago
cibuild.cmd	Fixed determinism and procdump path	5 months ago
clean_benchview_tools.cmd	Upload perf runs to Benchview (#14039)	a year ago
cleanup_perf.ps1	CR feedback	9 months ago
crossgen.sh	Quote variable/array expansions in shell scripts	2 months ago
deploy-msbuild.ps1	Make Powershell conform to our guidelines	7 months ago

Source: <https://github.com/dotnet/roslyn>

BUILD TOOLS



BUILD SERVERS



- ☛ Lock-in for complex scripts
- ☛ Hard to execute local
- ☛ Feedback cycle (occupying / money)
- ☛ Versioning?

BUILD SYSTEM

- 🕒 Tool orchestration
- 🕒 Build everywhere
- 🕒 Better feedback cycle
- 🕒 Minimal coupling
- 🕒 Versioning & Branching

GOTTA USE 'EM ALL!

BUILD SYSTEMS



MSBuild



FAKE



CAKE



PSAKE



RAKE



BUILD SYSTEMS



MSBuild



FAKE



CAKE



PSAKE



RAKE



Gradle

- ☛ Is MSBuild a handy build system?
- ☛ Yes, but mostly for project related information (precompile, code-generation).
- ☛ Not so much for tool invocations, loops, conditional code

PATH CONCATENATION

```
<CombinePath BasePath="$(OutDir)"
              Paths="$(ManifestFileName)"
              Condition="$(CreateModule) == 'true'">
  <Output TaskParameter="CombinedPaths"
          ItemName="ManifestPath"/>
</CombinePath>
```


BUILD SYSTEMS



- ☛ If you're using one of these, there are 3 different situations I can think of:
- ☛ 1. You're already using the language inside your project - everything alright.
- ☛ 2. You point out that one person, that is responsible for the build - might become a bottleneck.
- ☛ 3. Everyone needs to know about the new language - increased complexity for the team.

BUILD SYSTEMS



MSBuild



FAKE



CAKE



PSAKE



RAKE



Gradle

CAKE

```
var target = Argument("target", "Default");

Task("Default")
    .Does(() =>
    {
        Information("Hello from CAKE");
    });

RunTarget(target);
```

CAKE

```
var target = Argument("target", "Default");  
  
Task("Default")  
    .IsDependentOn("Initialization");  
  
Task("Initialisation");  
  
RunTarget(target);
```

Throws exception!

CAKE

```
Task( "Default" )  
    .Does( () =>  
{  
    Information("Hello from CAKE");  
    Just.Say("Something");  
});  
  
class Just  
{  
    public static void Say (string str)  
    {  
        Information(str);  
    }  
}
```

Doesn't compile!

CAKE

```
Task("Default")
    .Does(() =>
    {
        Information("Hello from CAKE");
        Just.Say("Something", Context);
    });

class Just
{
    public static void Say (string str, ICakeContext ctx)
    {
        ctx.Information(str);
    }
}
```



favs



- 🔔 Just an observation: experience is much better in VSCode
- 🔔 Anyone using Cake in other IDEs? How do you know which alias to use?

BUILD SYSTEMS



MSBuild



FAKE



CAKE



PSAKE



RAKE



Gradle



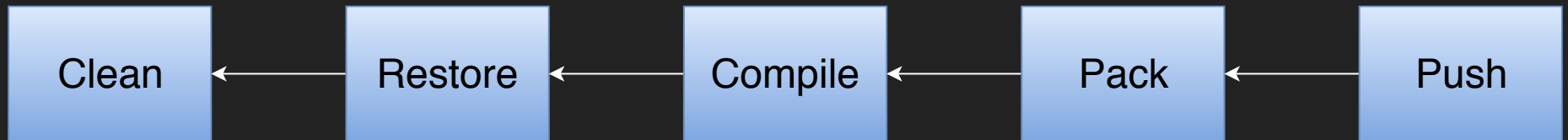
BUILD AUTOMATION SYSTEM FOR C# DEVELOPERS

SETUP & EXECUTION

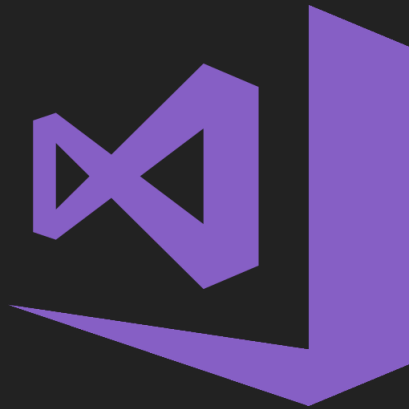
- ☛ Invoke setup script
- ☛ Tell about solution and platform choice
- ☛ `git status --untracked`
- ☛ Show added project in solution - it's a project!
- ☛ Help +

IMPLEMENTATION

- Console application, default target
- Targets are actual symbols -> refactoring/navigation



- ☛ Clean: predefined directories; globbing
- ☛ Restore: default settings
- ☛ Compile: GitVersion field
- ☛ Pack: path construction
- ☛ Push: parameter resolution



CONSOLE APP

- ☛ Want to discover a method? Go to declaration!
- ☛ Refactorings? Use your IDE features!
- ☛ Debugging? Set a breakpoint and run!

TOOLS



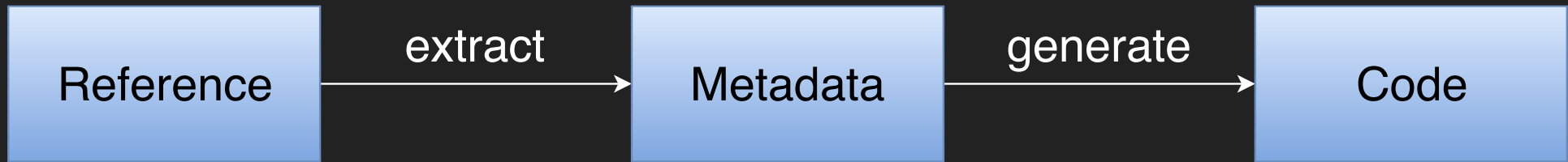
... and many more!

HOW DO WE GET SUPPORT?

A black cat with white paws is sitting on a wooden table, typing on a silver laptop keyboard. The cat's head is turned towards the camera, and its front paws are visible on the keys. The laptop is open, and the Apple logo is visible on the back of the lid. In the background, there is a wooden countertop with various items, including a red container and some glassware. The scene is set in a kitchen or dining area.

CODE-GENERATION

CODE-GENERATION

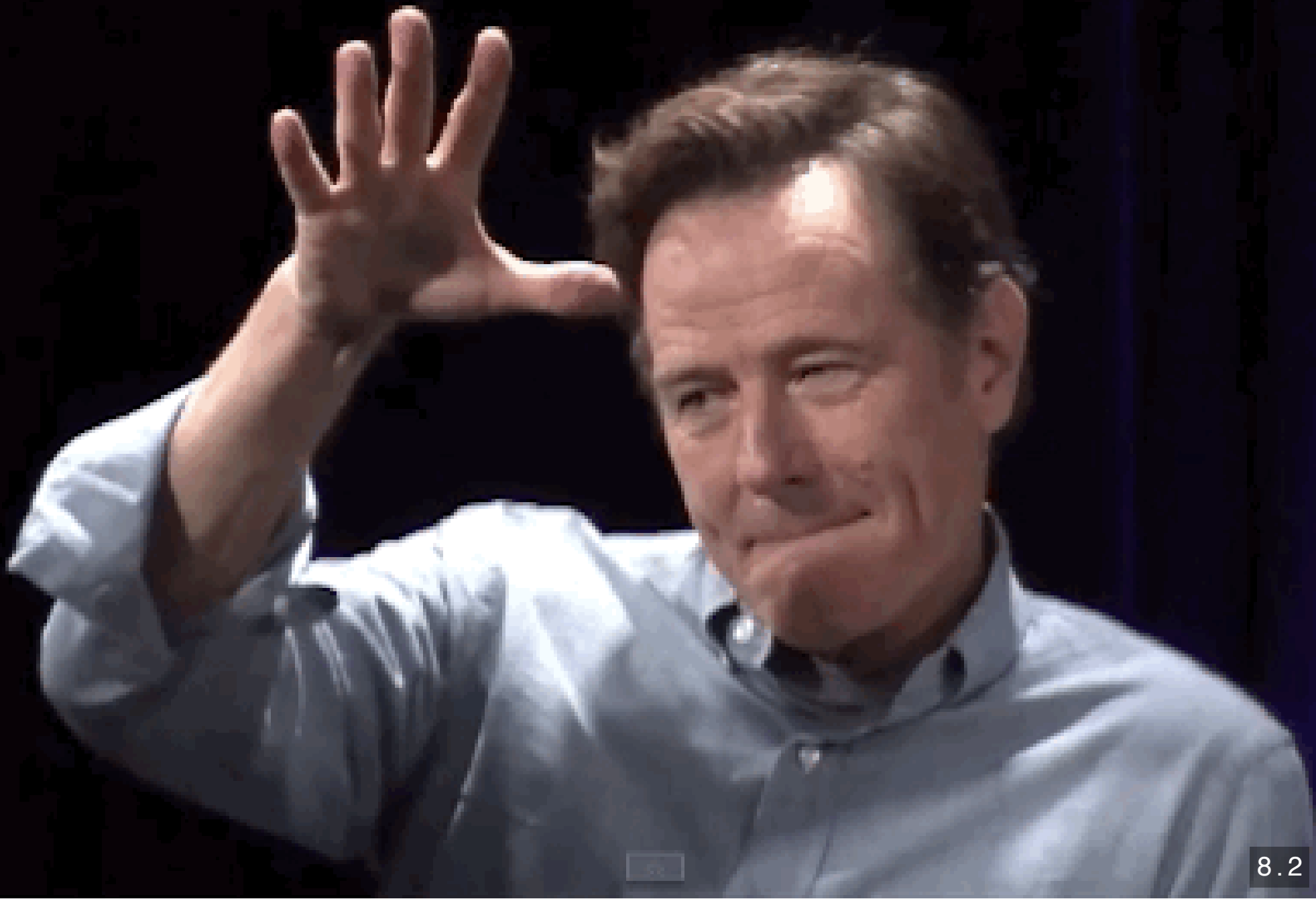


GENERATED CODE

- Invocation
- Tool resolution
- Argument configuration
- XML documentation
- Assertions

THANKS TO

- 👤 Patrick Schwarz
- 👤 Robert Schili
- 👤 Sebastian Karasek
- 👤 Stefan Bertels
- 👤 Ulrich Buchgraber





www.nuke.build



[nuke-build/nuke](https://github.com/nuke-build/nuke)



[@nukebuildnet](https://twitter.com/nukebuildnet)



gitter.im/nuke-build/nuke



present.nuke.build